

Attribute Database Management

นักเรียน

Entity ปกติ : Entity ที่สนใจและต้องการจัดเก็บข้อมูล

กรอบครัวลูกจ้าง

Entity อ่อนแอ : Entity ที่ขึ้นอยู่กับการมีอยู่ของ Entity อื่น (เรียกว่า) Parent Entity และมี Primary key เกิดจากส่วนหนึ่งหรือทั้งหมดของ Primary key ของ Parent Entity

รหัสประจำตัวนักเรียน

Primary key attribute : Attribute ที่เป็นคีย์หลัก

ชื่อ

นามสกุล

Simple attribute : Attribute ที่ไม่สามารถแบ่งย่อยได้อีก

ถนน

ตำบล

อำเภอ

Composite attribute : Attribute ที่สามารถแบ่งย่อยได้อีก

เลขที่บ้าน

ที่อยู่

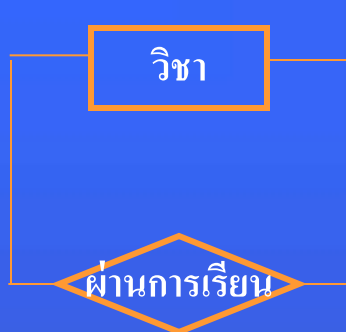
จังหวัด

ปริญญา

Multivalue attribute : Attribute ที่สามารถมีได้หลายค่า

อายุ

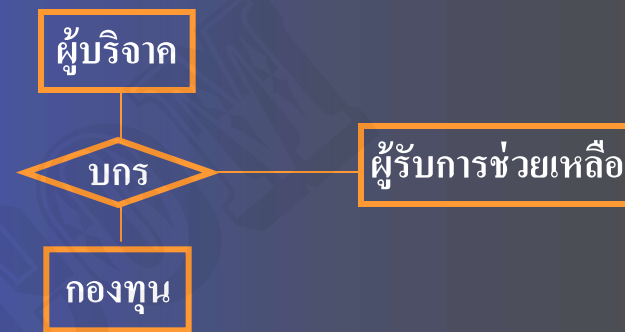
Derive attribute : Attribute ที่สามารถหาค่ามาได้จากการคำนวณจาก Attribute อื่น ที่มีอยู่

Relationship : ระดับความสัมพันธ์

Unary relationship



Binary relationship



Ternary relationship

Relationship : ลักษณะต่าง ๆ ที่เกี่ยวกับความสัมพันธ์

Cardinality : ระบุค่าสูงสุด ต่ำสุดใน Entity หนึ่งที่มีความสัมพันธ์กับ Entity หนึ่ง

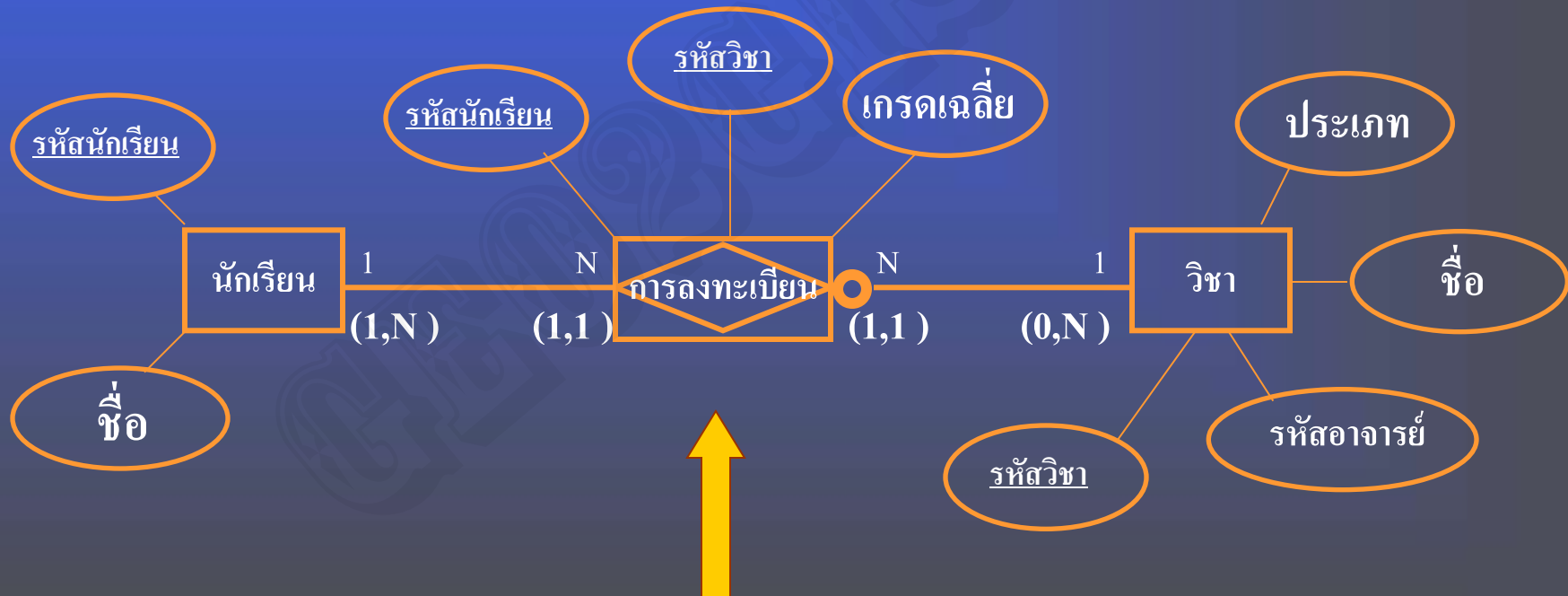


ความสัมพันธ์แบบเลือก(Optional) และแบบบังคับ(Mandatory)



Relationship : ลักษณะต่าง ๆ ที่เกี่ยวกับความสัมพันธ์ Attribute Database Management

Composite entity : Entity ที่สร้างขึ้นเพื่อแปลงสัมพันธ์แบบ $M : N$ ให้เป็น $1 : N$



Composite entity

กระบวนการออกแบบ E-R model

- กำหนด Entity ที่ต้องการในระบบ
- กำหนดความสัมพันธ์ระหว่าง Entity
- กำหนด Attribute ให้กับ Entity

กระบวนการออกแบบ E-R model

ตัวอย่างการสร้าง E-R model

1. บริษัท ที่ปรึกษา A มี 5 แผนก (Division)
2. แต่ละแผนก ทำโครงการ (Project) ได้หลายโครงการ
3. แต่ละโครงการขึ้นอยู่กับแผนกเดียว
4. แต่ละแผนกมีพนักงาน (Employee) ได้หลายคน พนักงานแต่ละคนสังกัดอยู่แผนก

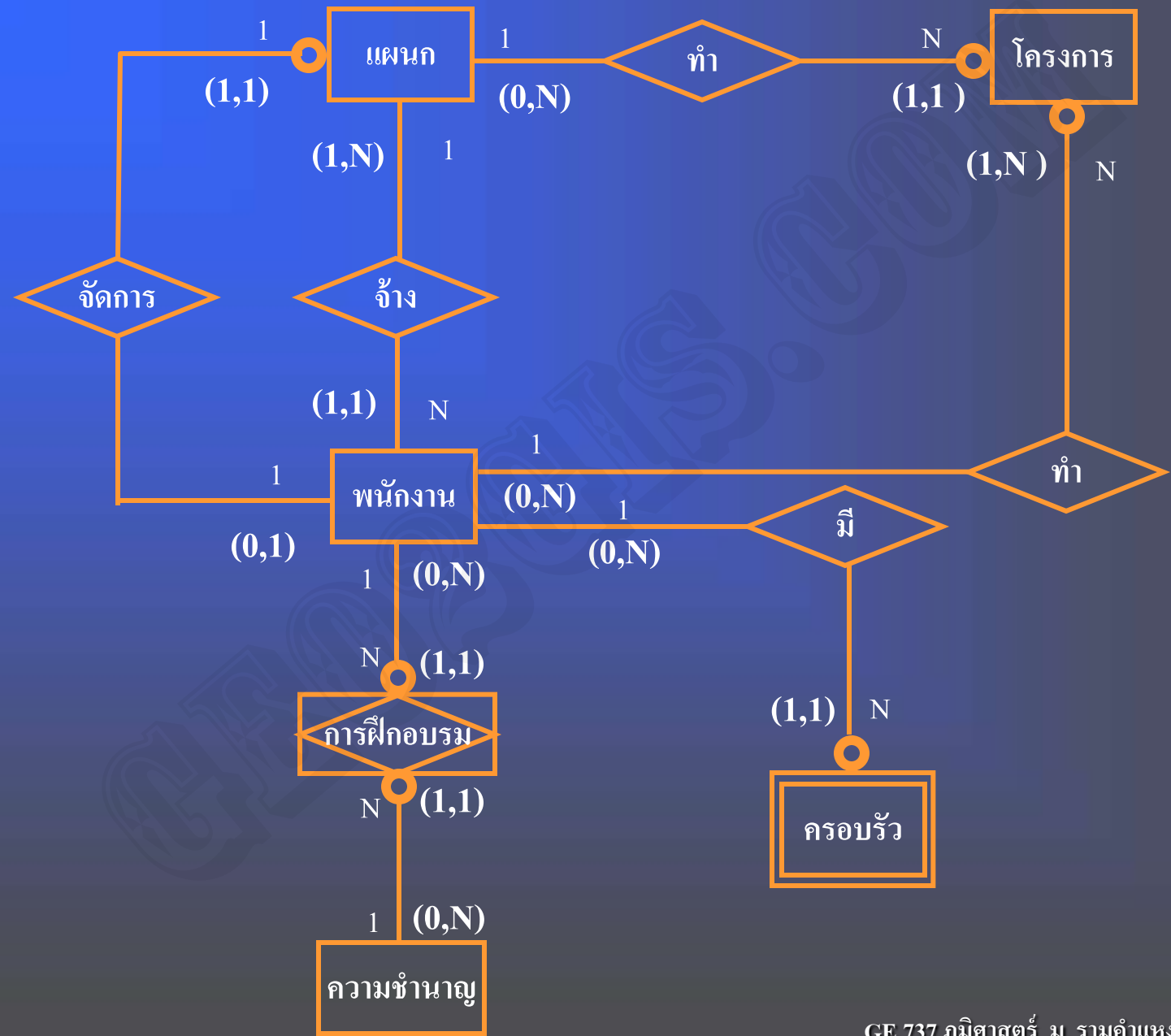
ใดแผนกหนึ่งเท่านั้น

5. แต่ละแผนกมีพนักงาน 1 คนเป็นผู้จัดการ
6. พนักงานอาจจะมีหรือไม่มีครอบครัว (Family) ก็ได้
7. พนักงานแต่ละคนทำงานได้หลายโครงการหรือไม่ทำเลยก็ได้
8. แต่ละโครงการต้องมีพนักงานอย่างน้อย 1 คนทำงาน
9. บริษัทฯ รวบรวมความชำนาญ (Skill) ของพนักงานแต่ละคนไว้
10. ความชำนาญอย่างหนึ่งอาจมีพนักงานหลายคนทำได้
11. พนักงานคนหนึ่งอาจจะชำนาญหลายอย่าง

ตัวอย่างการสร้าง E-R model

<u>Entity</u>	<u>Relationship</u>	<u>Connectivity</u>	<u>Entity</u>
แผนก	ทำ	1 : N	โครงการ
แผนก	มี	1 : N	พนักงาน
แผนก	จัดการ	1 : 1	พนักงาน
พนักงาน	มี	1 : 1	ครอบครัว
พนักงาน	ทำ	1 : N	โครงการ
พนักงาน	มี	M : N	ความชำนาญ

ตัวอย่างการสร้าง E-R model



การแปลง E-R Diagram ให้เป็นโครงสร้างฐานข้อมูล

สร้างตารางจาก E-R Diagram

1. ใช้ชื่อของ Entity เป็นชื่อของตาราง จากนั้นกำหนด Attribute พร้อมกำหนด Primary key
2. กรณี Relationship ของ Entity เป็นแบบ $1 : N$ ให้นำ Primary key ของ Entity ที่อยู่ข้าง 1 ไปใส่เป็น Foreign key ในตารางที่อยู่ข้าง N
3. ในกรณีของ Entity อ่อนแอ ให้ใช้ Primary key ของ Parent Entity เป็น Primary key ของ Entity อ่อนแอ หรือส่วนหนึ่งของ Primary key ของ Entity อ่อนแอ

E-R model เป็นเครื่องมือในการออกแบบระบบฐานข้อมูลเชิงแนวคิด นำเสนอรายละเอียดให้ผู้ใช้และผู้ออกแบบเห็นภาพรวมฐานข้อมูลทั้งระบบ และเห็นความสัมพันธ์ของข้อมูลอย่างชัดเจน

กระบวนการ Normalization

- กระบวนการวิเคราะห์ความสัมพันธ์ระหว่าง Attribute ในแต่ละ Relationship เพื่อตรวจสอบและแก้ไขปัญหาคำซ้ำซ้อนของข้อมูลในแต่ละ Relation ให้เหลือน้อยที่สุด หรือจนกระทั่งไม่มีความซ้ำซ้อนอยู่เลย
- การที่ต้องมีกระบวนการ Normalization เนื่องจากขั้นตอนการออกแบบโดยใช้ E-R model เป็นการวิเคราะห์ความสัมพันธ์ระดับ Entity เท่านั้น แต่ไม่สามารถวิเคราะห์ความสัมพันธ์ระหว่าง Attribute ในแต่ละ Relationship ได้

Comp_ID	Company	City	Region	Product_ID	Prod_Name	Color
C01	เอ เฟอร์นิเจอร์	กทม.	กลาง	PD01	เก้าอี้	แดง
C01	เอ เฟอร์นิเจอร์	กทม.	กลาง	PD02	เก้าอี้	น้ำตาล
C02	บี เฟอร์นิเจอร์	เลย	อีสาน	PD11	โต๊ะ	น้ำตาล
C02	บี เฟอร์นิเจอร์	เลย	อีสาน	PD12	โต๊ะ	แดง
C02	บี เฟอร์นิเจอร์	เลย	อีสาน	PD01	เก้าอี้	แดง
C03	C Furniture	กทม.	กลาง	PD02	เก้าอี้	น้ำตาล
C04	D Furniture	ตาก	เหนือ	PD11	โต๊ะ	น้ำตาล
C04	D Furniture	ตาก	เหนือ	PD03	เก้าอี้	ฟ้า
C04	D Furniture	ตาก	เหนือ	PD21	ตู้เหล็ก	แดง

จัดทำตารางข้อมูลให้มีคุณสมบัติอยู่ใน Normal Form แบบต่าง ๆ

กระบวนการวิเคราะห์ความสัมพันธ์ระหว่าง Attribute ในแต่ละ Relationship

เพื่อตรวจสอบและแก้ไขปัญหาค่าซ้ำซ้อนของข้อมูลในแต่ละ Relation

ให้เหลือน้อยที่สุด หรือจนกระทั่งไม่มีความซ้ำซ้อนอยู่เลย

1. Normal Form ที่ 1 หรือ 1NF
2. Normal Form ที่ 2 หรือ 2NF
3. Normal Form ที่ 3 หรือ 3NF
4. Normal Form แบบบอยซ์คอดด์ หรือ BCNF (Boyce-Codd Normal Form)
5. Normal Form ที่ 4 หรือ 4NF
6. Normal Form ที่ 5 หรือ 5NF

การขึ้นต่อกันแบบฟังก์ชัน

(Functional Dependency : FD)

ความหมาย : $f(x) = x + 1$

$y = x + 1$

ความสัมพันธ์ของ Attribute ในตารางใด ๆ ในรูปแบบฟังก์ชัน โดยที่
หากรู้ค่า Attribute หนึ่ง จะทำให้รู้ค่า Attribute อื่นในแถวเดียวกัน

City	Region
กทม.	กลาง
เลย	อีสาน
ตาก	เหนือ

ตัวอย่าง

เมื่อทราบค่าใน Attribute City จะทำให้ทราบค่าใน Attribute Region

เรียก Attribute City ว่า “Determinant”

Attribute Region ว่า “Dependent”

สัญลักษณ์ City $\longrightarrow$ Region หรือ



อ่านว่า City Determine Region หรือ Region ขึ้นต่อกันแบบฟังก์ชันกับ City

ตัวอย่าง

การขึ้นต่อกันแบบฟังก์ชัน (FD) :

Determinant , Dependent มีเพียง 1 Attribute

Stu_ID	F_Name
490101	อรุณ
490102	วิชัย
490103	กำพล

Stu_ID $\longrightarrow$ F_Name หรือ F_Name $\longrightarrow$ Stu_ID หรือ

F_Name $\longleftrightarrow$ Stu_ID

Determinant มากกว่า 1 Attribute , Dependent มีเพียง 1 Attribute

Stu_ID	Major	Credit
490101	Geo	45
490102	Math	77
490103	Geo	55

Stu_ID, Major $\longrightarrow$ Credit



การขึ้นต่อกันแบบฟังก์ชัน (FD) :

Determinant มีเพียง 1 Attribute, **Dependent** มากกว่า 1 Attribute

Stu_ID	Credit	GPA	Major	Date
490101	45	2.5	Geo	11/01/49
490102	77	3.3	Math	11/01/49
490103	55	2.4	Geo	15/01/49

Stu_ID $\longrightarrow$ Credit, GPA, Major, Date

หรือ

Stu_ID $\longrightarrow$ Credit

Stu_ID $\longrightarrow$ GPA

Stu_ID $\longrightarrow$ Major

Stu_ID $\longrightarrow$ Date



รูปแบบการขึ้นต่อกัน

1. การขึ้นต่อกันแบบฟังก์ชัน (Functional Dependency : FD)
 - 1.1 การขึ้นต่อกันแบบทั้งหมด (Fully Functional Dependency)
 - 1.2 การขึ้นต่อกันแบบบางส่วน (Partial Dependency)
 - 1.3 การขึ้นต่อกันแบบ Transitive (Transitive Dependency)
2. การขึ้นต่อกันแบบเชิงกลุ่ม (Multi Value Dependency : MVD)
3. การขึ้นต่อกันแบบ Join (Join Dependency : JD)

รูปแบบการขึ้นต่อกัน

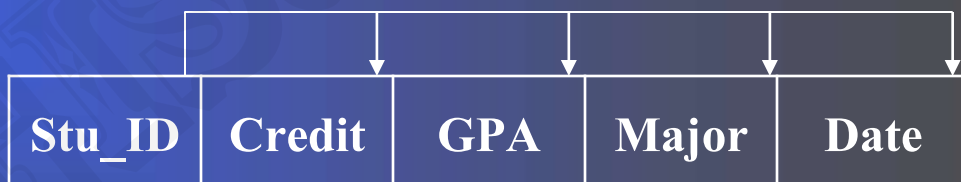
การขึ้นต่อกันแบบฟังก์ชัน (Functional Dependency : FD)

การขึ้นต่อกันแบบทั้งหมด (Fully Functional Dependency)

- Determinant มีเพียง **1** Attribute

Stu_ID	Credit	GPA	Major	Date
490101	45	2.5	Geo	11/01/49
490102	77	3.3	Math	11/01/49
490103	55	2.4	Geo	15/01/49

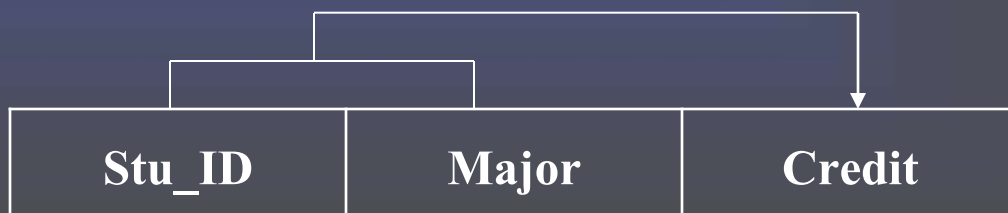
FD1: Stu_ID $\longrightarrow$ Credit, GPA, Major, Date



- Determinant มีมากกว่า **1** Attribute

Stu_ID	Major	Credit
490101	Geo	45
490102	Math	77
490103	Geo	55

FD2: Stu_ID, Major $\longrightarrow$ Credit



รูปแบบการขึ้นต่อกัน

การขึ้นต่อกันแบบฟังก์ชัน (Functional Dependency : FD)

การขึ้นต่อกันแบบบางส่วน (Partial Dependency)

Stu_ID	Maj_ID	Major	Credit
490101	M01	Geo	45
490201	M05	Math	77
490102	M01	Geo	55
490202	M05	Math	33
490303	M02	Social	41

รู้ค่าใดค่าหนึ่งของ Attribute ที่เป็น Determinant (เป็น Key Attribute) เพียงพอที่จะรู้ค่าของ Attribute ที่เป็น Dependent

การขึ้นต่อกันแบบฟังก์ชันที่มี

Attribute ที่เป็น Dependent ไม่ได้ขึ้นอยู่กับ Attribute ที่เป็น Determinant

FD3: Stu_ID, Maj_ID $\longrightarrow$ Major, Credit

FD4: Maj_ID $\longrightarrow$ Major



รูปแบบการขึ้นต่อกัน

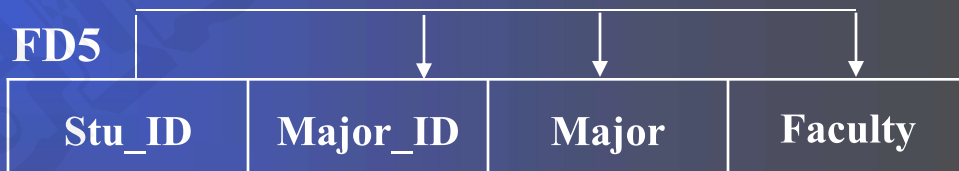
การขึ้นต่อกันแบบฟังก์ชัน (Functional Dependency : FD)

การขึ้นต่อกันแบบ Transitive (Transitive Dependency)

Stu_ID	Maj_ID	Major	Faculty
490101	M01	Geo	ศึกษาศาสตร์
490201	M05	Math	วิทยาศาสตร์
490102	M01	Geo	ศึกษาศาสตร์
490202	M05	Math	วิทยาศาสตร์
490303	M02	Social	สังคมศาสตร์

FD5: Stu_ID, Maj_ID $\longrightarrow$ Major, Faculty

FD6: Major $\longrightarrow$ Faculty



Attribute ที่ไม่ใช่ Key Attribute สามารถระบุค่าของ Attribute ที่ไม่ใช่ Key Attribute

ได้

รูปแบบการขึ้นต่อกัน

การขึ้นต่อกันแบบเชิงกลุ่ม (Multi Value Dependency : MVD)

ความสัมพันธ์ตั้งแต่ 3 Attribute ขึ้นไป โดย Attribute ที่เป็น Determinant สามารถระบุค่า Attribute ที่เป็น Dependent ได้หลายค่า

การนำ Attribute ทั้ง 3 มาไว้ในตารางเดียวกัน การขึ้นต่อกันแบบเชิงกลุ่มจะเกิดดังนี้
 เมื่อ Attribute X, Y, Z อยู่ในตารางเดียวกัน โดยที่ Attribute X สัมพันธ์กับ Attribute Y และ Z แต่ Attribute Y และ Z ไม่มีส่วนเกี่ยวข้องกันเลย

$X \longrightarrow Y$ อ่านว่า X multidetermine Y หมายความว่า
 เมื่อทราบค่า X 1 ค่า หรือ มากกว่า 1 ค่า จะทำให้รู้ค่า Y หลายค่า
 (เป็นความสัมพันธ์แบบ 1 : M หรือ M : N)

$X \longrightarrow Z$ อ่านว่า X multidetermine Z หมายความว่า
 เมื่อทราบค่า X 1 ค่า หรือ มากกว่า 1 ค่า จะทำให้รู้ค่า Z หลายค่า

Y และ Z ไม่มีความสัมพันธ์กัน

รูปแบบการขึ้นต่อกัน

การขึ้นต่อกันแบบเชิงกลุ่ม (Multi Value Dependency : MVD)

Emp_ID	Proj_ID	Language
E01	Po1	อังกฤษ
	Po2	จีน
E02	Po1	ฝรั่งเศส ญี่ปุ่น เยอรมัน
	Po3	ฝรั่งเศส

Emp_ID	Proj_ID	Language
E01	Po1	อังกฤษ
E01	Po1	จีน
E01	Po2	อังกฤษ
E01	Po2	จีน
E02	Po1	ฝรั่งเศส
E02	Po1	ญี่ปุ่น
E02	Po1	เยอรมัน
E03	Po3	ฝรั่งเศส

เอา Attribute ที่ไม่มีความสัมพันธ์กัน (Proj_ID และ Language) มาอยู่ในตารางกัน โดยมีตัว Determinate ร่วมกัน (Proj_ID) ทำให้เกิดข้อมูลซ้ำกัน

Emp_ID $\longrightarrow$ Proj_ID เมื่อทราบค่า Emp_ID 1 ค่า หรือ มากกว่า 1 ค่า จะทำให้รู้ค่า Proj_ID หลายค่า

Emp_ID $\longrightarrow$ Language เมื่อทราบค่า Emp_ID 1 ค่า หรือ มากกว่า 1 ค่า จะทำให้รู้ค่า Language หลายค่า

Proj_ID และ Language ไม่มีความสัมพันธ์กัน

รูปแบบการขึ้นต่อกัน

การขึ้นต่อกันแบบ Join (Join Dependency : JD)

Comp_ID	Company	City	Region
C01	เอ เฟอร์นิเจอร์	กทม.	กลาง
C02	เอ เฟอร์นิเจอร์	เลย	อีสาน
C03	C Furniture	กทม.	กลาง
C04	D Furniture	ตาก	เหนือ

Comp_ID	City
C01	กทม.
C02	เลย
C03	กทม.
C04	ตาก

ถ้าแยกตารางออกเป็นตารางย่อย ๆ แล้ว
เมื่อนำเอามารวมใหม่ ต้องได้ตาราง
เหมือนตารางเดิม

กรณีเมื่อนำเอามารวมใหม่ แล้ว ได้
ตารางไม่เหมือนตารางเดิม เกิดขึ้น
2 กรณี คือ มีข้อมูลแถวใหม่ หรือ
ข้อมูลบางแถวหายไป

Comp_ID	Company
C01	เอ เฟอร์นิเจอร์
C02	เอ เฟอร์นิเจอร์
C03	C Furniture
C04	D Furniture

City	Region
กทม.	กลาง
เลย	อีสาน
ตาก	เหนือ

Comp_ID	Company	City	Region
C01	เอ เฟอร์นิเจอร์	กทม.	กลาง
C02	เอ เฟอร์นิเจอร์	เลย	อีสาน
C03	C Furniture	กทม.	กลาง
C04	D Furniture	ตาก	เหนือ

จัดทำตารางข้อมูลให้มีคุณสมบัติอยู่ในNormal Form ที่ 1 หรือ 1NFคุณสมบัติ

- ค่าใน Attribute แต่ละแถวเป็นค่าข้อมูลเดี่ยว ๆ
- ในตารางไม่มีข้อมูลที่เป็นกลุ่มซ้ำ

Comp_ID	Company	City	Region	Product_ID	Prod_Name	Color	Quantity
C01	เอ เฟอร์นิเจอร์	กทม.	กลาง	PD01	เก้าอี้	แดง	50
				PD02	เก้าอี้	น้ำตาล	20
C02	บี เฟอร์นิเจอร์	เลย	อีสาน	PD11	โต๊ะ	น้ำตาล	30
				PD12	โต๊ะ	แดง	25
				PD01	เก้าอี้	แดง	25
C03	C Furniture	กทม.	กลาง	PD02	เก้าอี้	น้ำตาล	30
C04	D Furniture	ตาก	เหนือ	PD11	โต๊ะ	น้ำตาล	35
				PD03	เก้าอี้	ฟ้า	40
				PD21	ตู้เหล็ก	แดง	30

ค่าใน Attribute
แต่ละแถวไม่เป็น
ค่าข้อมูลเดี่ยว ๆ

วิธีการ

ทำให้ทุก Attribute
มีค่าโดยเติมค่า
ในบรรทัดที่ 2 4 5 8 9

จัดทำตารางข้อมูลให้มีคุณสมบัติอยู่ในNormal Form ที่ 1 หรือ 1NF

1. ค่าใน Attribute แต่ละแถวเป็นค่าข้อมูลเดี่ยว ๆ
2. ในตารางไม่มีข้อมูลที่เป็นกลุ่มซ้ำ

Comp_ID	Company	City	Region	Product_ID	Prod_Name	Color	Quantity
C01	เอ เฟอร์นิเจอร์	กทม.	กลาง	PD01	เก้าอี้	แดง	50
C01	เอ เฟอร์นิเจอร์	กทม.	กลาง	PD02	เก้าอี้	น้ำตาล	20
C02	บี เฟอร์นิเจอร์	เลย	อีสาน	PD11	โต๊ะ	น้ำตาล	30
C02	บี เฟอร์นิเจอร์	เลย	อีสาน	PD12	โต๊ะ	แดง	25
C02	บี เฟอร์นิเจอร์	เลย	อีสาน	PD01	เก้าอี้	แดง	25
C03	C Furniture	กทม.	กลาง	PD02	เก้าอี้	น้ำตาล	30
C04	D Furniture	ตาก	เหนือ	PD11	โต๊ะ	น้ำตาล	35
C04	D Furniture	ตาก	เหนือ	PD03	เก้าอี้	ฟ้า	40
C04	D Furniture	ตาก	เหนือ	PD21	ตู้เหล็ก	แดง	30

ค่าใน Attribute
แต่ละแถวเป็น
ค่าข้อมูลเดี่ยว ๆ

ในตาราง
มีข้อมูล
ที่เป็นกลุ่มซ้ำ

จัดทำตารางข้อมูลให้มีคุณสมบัติอยู่ในNormal Form ที่ 1 หรือ 1NF

Company

Comp_ID	Company	City	Region
C01	เอ เฟอร์นิเจอร์	กทม.	กลาง
C02	บี เฟอร์นิเจอร์	เลย	อีสาน
C03	C Furniture	กทม.	กลาง
C04	D Furniture	ตาก	เหนือ

Company_product

Comp_ID	Product_ID	Prod_Name	Color	Quantity
C01	PD01	เก้าอี้	แดง	50
C01	PD02	เก้าอี้	น้ำตาล	20
C02	PD11	โต๊ะ	น้ำตาล	30
C02	PD12	โต๊ะ	แดง	25
C02	PD01	เก้าอี้	แดง	25
C03	PD02	เก้าอี้	น้ำตาล	30
C04	PD11	โต๊ะ	น้ำตาล	35
C04	PD03	เก้าอี้	ฟ้า	40
C04	PD21	ตู้เหล็ก	แดง	30

วิธีการ

แยกตารางออกเป็นตารางย่อย ๆ

เงื่อนไข เมื่อนำเอามารวมใหม่

ต้องได้ตารางเหมือนตารางเดิม

ควรนำ Attribute ที่มีความสัมพันธ์
กันไว้ตารางเดียวกัน

ทั้ง 2 ตารางมีคุณสมบัติอยู่ใน 1NF

จัดทำตารางข้อมูลให้มีคุณสมบัติอยู่ใน

Normal Form ที่ 2 หรือ 2NF

1NF ยังไม่มีการวิเคราะห์ความสัมพันธ์ระหว่าง Attribute

คุณสมบัติ

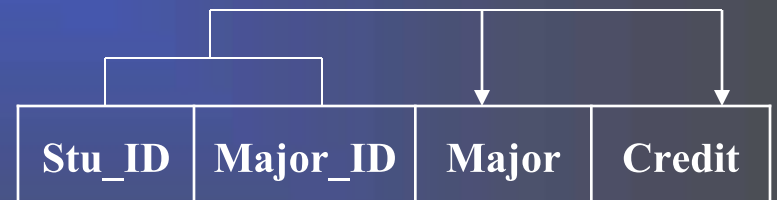
- ตารางนั้นต้องมีคุณสมบัติ 1NF
- Attribute ที่ไม่ใช่ Key ต้องขึ้นกับ Key หลักอย่างแท้จริง (ไม่มีการขึ้นต่อกันแบบบางส่วน)



เมื่อ Stu_ID = Key หลัก



เมื่อ Stu_ID และ Major = Key หลัก



เมื่อ Stu_ID และ Major = Key หลัก

จัดทำตารางข้อมูลให้มีคุณสมบัติอยู่ใน Normal Form ที่ 2 หรือ 2NF

Company

Comp_ID	Company	City	Region
C01	เอ เฟอร์นิเจอร์	กทม.	กลาง
C02	บี เฟอร์นิเจอร์	เลย	อีสาน
C03	C Furniture	กทม.	กลาง
C04	D Furniture	ตาก	เหนือ

ตรวจสอบคุณสมบัติ

Attribute ที่ไม่ใช่ Key ต้องขึ้นกับ

Key หลักอย่างแท้จริง

(**ไม่**มีการขึ้นต่อกันแบบบางส่วน)

ทั้ง 2 ตารางมีคุณสมบัติอยู่ใน 1NF

Company_product

Comp_ID	Product_ID	Prod_Name	Color	Quantity
C01	PD01	เก้าอี้	แดง	50
C01	PD02	เก้าอี้	น้ำตาล	20
C02	PD11	โต๊ะ	น้ำตาล	30
C02	PD12	โต๊ะ	แดง	25
C02	PD01	เก้าอี้	แดง	25
C03	PD02	เก้าอี้	น้ำตาล	30
C04	PD11	โต๊ะ	น้ำตาล	35
C04	PD03	เก้าอี้	ฟ้า	40
C04	PD21	ตู้เหล็ก	แดง	30

จัดทำตารางข้อมูลให้มีคุณสมบัติอยู่ในNormal Form ที่ 2 หรือ 2NF

Company

Comp_ID	Company	City	Region
C01	เอ เฟอร์นิเจอร์	กทม.	กลาง
C02	บี เฟอร์นิเจอร์	เลย	อีสาน
C03	C Furniture	กทม.	กลาง
C04	D Furniture	ตาก	เหนือ

ตรวจสอบคุณสมบัติ

Attribute ที่ไม่ใช่ Key ต้องขึ้นกับ
Key หลักอย่างแท้จริง
(**ไม่**มีการขึ้นต่อกันแบบบางส่วน)

วิธีการ

ตรวจสอบคุณสมบัติของตาราง Company

Attribute ที่เป็น Key หลัก คือ **Comp_ID**

Attribute ที่ไม่ใช่ Key คือ **Company, City, Region**

FD: **Comp_ID** $\longrightarrow$ **Company, City, Region**

ตาราง Company มีคุณสมบัติอยู่ใน 2NF

จัดทำตารางข้อมูลให้มีคุณสมบัติอยู่ใน

Normal Form ที่ 2 หรือ 2NF

ตรวจสอบคุณสมบัติ

Attribute ที่ไม่ใช่ Key ต้องขึ้นกับ

Key หลักอย่างแท้จริง

(ไม่มีการขึ้นต่อกันแบบบางส่วน)

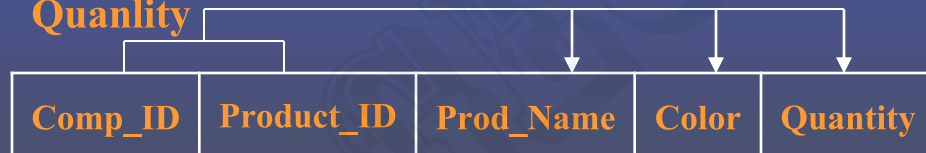
วิธีการ

ตรวจสอบคุณสมบัติของตาราง Company_product

Attribute ที่เป็น Key หลัก คือ **Comp_ID, Product_ID**

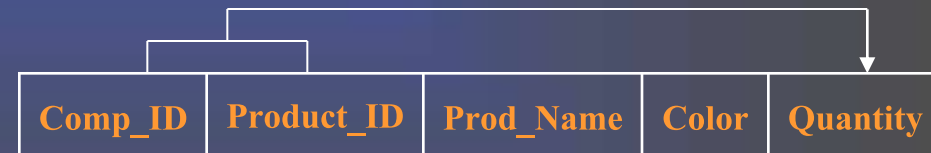
Attribute ที่ไม่ใช่ Key คือ **Prod_Name, Color,**

Quantity



Company_product

Comp_ID	Product_ID	Prod_Name	Color	Quantity
C01	PD01	เก้าอี้	แดง	50
C01	PD02	เก้าอี้	น้ำตาล	20
C02	PD11	โต๊ะ	น้ำตาล	30
C02	PD12	โต๊ะ	แดง	25
C02	PD01	เก้าอี้	แดง	25
C03	PD02	เก้าอี้	น้ำตาล	30
C04	PD11	โต๊ะ	น้ำตาล	35
C04	PD03	เก้าอี้	ฟ้า	40
C04	PD21	ตู้เหล็ก	แดง	30



มีการขึ้นต่อกันแบบบางส่วน

ตาราง Company ยังไม่มีคุณสมบัติอยู่ใน 2NF

จัดทำตารางข้อมูลให้มีคุณสมบัติอยู่ใน

Normal Form ที่ 2 หรือ 2NF

ตรวจสอบคุณสมบัติ

Attribute ที่ไม่ใช่ Key ต้องขึ้นกับ Key หลัก
อย่างแท้จริง (ไม่มีการขึ้นต่อกันแบบบางส่วน)

วิธีการ

แยกตารางใหม่ให้มีคุณสมบัติอยู่ใน 2NF

Company_product

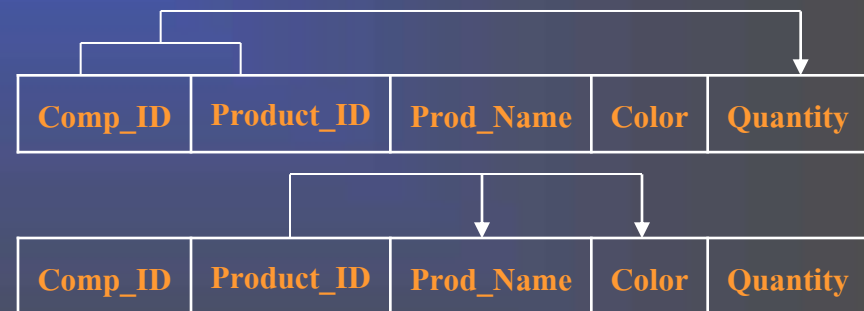
Comp_ID	Product_ID	Quantity
C01	PD01	50
C01	PD02	20
C02	PD11	30
C02	PD12	25
C02	PD01	25
C03	PD02	30
C04	PD11	35
C04	PD03	40
C04	PD21	30

Product

Product_ID	Prod_Name	Color
PD01	เก้าอี้	แดง
PD02	เก้าอี้	น้ำตาล
PD11	โต๊ะ	น้ำตาล
PD12	โต๊ะ	แดง
PD03	เก้าอี้	ฟ้า
PD21	ตู้เหล็ก	แดง

Attribute Database Management
Company_product

Comp_ID	Product_ID	Prod_Name	Color	Quantity
C01	PD01	เก้าอี้	แดง	50
C01	PD02	เก้าอี้	น้ำตาล	20
C02	PD11	โต๊ะ	น้ำตาล	30
C02	PD12	โต๊ะ	แดง	25
C02	PD01	เก้าอี้	แดง	25
C03	PD02	เก้าอี้	น้ำตาล	30
C04	PD11	โต๊ะ	น้ำตาล	35
C04	PD03	เก้าอี้	ฟ้า	40
C04	PD21	ตู้เหล็ก	แดง	30



ทั้ง 2 ตาราง มีคุณสมบัติอยู่ใน 2NF

จัดทำตารางข้อมูลให้มีคุณสมบัติอยู่ใน

Normal Form ที่ 3 หรือ 3NF

คุณสมบัติ

- ตารางนั้นต้องมีคุณสมบัติ 2NF
- Attribute ในตารางไม่มีการขึ้นต่อกันแบบ Transitive



เมื่อ Stu_ID และ Major = Key หลัก

จัดทำตารางข้อมูลให้มีคุณสมบัติอยู่ในNormal Form ที่ 3 หรือ 3NFตรวจสอบคุณสมบัติมีคุณสมบัติอยู่ใน 2NF

Company_product

Comp_ID	Product_ID	Quantity
C01	PD01	50
C01	PD02	20
C02	PD11	30
C02	PD12	25
C02	PD01	25
C03	PD02	30
C04	PD11	35
C04	PD03	40
C04	PD21	30

Product

Product_ID	Prod_Name	Color
PD01	เก้าอี้	แดง
PD02	เก้าอี้	น้ำตาล
PD11	โต๊ะ	น้ำตาล
PD12	โต๊ะ	แดง
PD03	เก้าอี้	ฟ้า
PD21	ตู้เหล็ก	แดง

Company

Comp_ID	Company	City	Region
C01	เอ เฟอร์นิเจอร์	กทม.	กลาง
C02	บี เฟอร์นิเจอร์	เลย	อีสาน
C03	C Furniture	กทม.	กลาง
C04	D Furniture	ตาก	เหนือ

ทั้ง 3 ตาราง
มีคุณสมบัติอยู่ใน 2NF

จัดทำตารางข้อมูลให้มีคุณสมบัติอยู่ใน

Normal Form ที่ 3 หรือ 3NF

ตรวจสอบคุณสมบัติ

ไม่มีการขึ้นต่อกันแบบ Transitive

Company_product

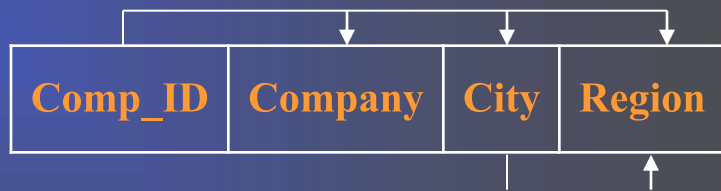
Comp_ID	Product_ID	Quantity
C01	PD01	50
C01	PD02	20
C02	PD11	30
C02	PD12	25
C02	PD01	25
C03	PD02	30
C04	PD11	35
C04	PD03	40
C04	PD21	30

Product

Product_ID	Prod_Name	Color
PD01	เก้าอี้	แดง
PD02	เก้าอี้	น้ำตาล
PD11	โต๊ะ	น้ำตาล
PD12	โต๊ะ	แดง
PD03	เก้าอี้	ฟ้า
PD21	ตู้เหล็ก	แดง

Company

Comp_ID	Company	City	Region
C01	เอ เฟอร์นิเจอร์	กทม.	กลาง
C02	บี เฟอร์นิเจอร์	เลย	อีสาน
C03	C Furniture	กทม.	กลาง
C04	D Furniture	ตาก	เหนือ



ตาราง Company_product และ Product ไม่มีการขึ้นต่อกันแบบ Transitive (มีคุณสมบัติอยู่ใน 3NF) แต่ ตาราง Company มีการขึ้นต่อกันแบบ Transitive (ไม่มีคุณสมบัติอยู่ใน 3NF)

จัดทำตารางข้อมูลให้มีคุณสมบัติอยู่ในNormal Form ที่ 3 หรือ 3NFตรวจสอบคุณสมบัติไม่มีการขึ้นต่อกันแบบ Transitiveวิธีการ

กำจัดการขึ้นต่อกันแบบ Transitive ออกจาก
ตาราง Company โดยแยก Attribute ที่ทำให้เกิด
การขึ้นต่อกันแบบ Transitive ออกเป็น 2 ตาราง

แยกตารางใหม่ คือ Zone โดยกำหนด
Key หลักใหม่ คือ City ส่วนตาราง
Company ให้คง Attribute City ไว้
เพื่อเป็น Foreign key และ ตาราง
Company และ Zone ไม่มีการขึ้นต่อ
กันแบบ Transitive **จึงมีคุณสมบัติอยู่
ใน 3NF**

Company

Comp_ID	Company	City	Region
C01	เอ เฟอร์นิเจอร์	กทม.	กลาง
C02	บี เฟอร์นิเจอร์	เลย	อีสาน
C03	C Furniture	กทม.	กลาง
C04	D Furniture	ตาก	เหนือ

Company

Comp_ID	Company	City
C01	เอ เฟอร์นิเจอร์	กทม.
C02	บี เฟอร์นิเจอร์	เลย
C03	C Furniture	กทม.
C04	D Furniture	ตาก

Zone

City	Region
กทม.	กลาง
เลย	อีสาน
ตาก	เหนือ

คีย์คู่แข่ง (Candidate key) : Attribute ตัวหนึ่ง หรือหลายตัวรวมกัน ซึ่งเกิดค่า Unique

โดยไม่นำ Attribute ตัวหนึ่งซึ่งเป็นค่า Unique แล้วมารวมด้วย

Stu_ID	F_Name	L_Name	Pers_ID	Credit	GPA	Date	Phone
490101	อรุณ	สมบัติ	10021110101	45	2.5	11/01/49	025216447
490102	วิชัย	เสรีไทย	15021110201	77	3.3	11/01/49	025387722
490103	กำพล	ดินสอ	28021510507	55	2.4	15/01/49	024126333

Stu_ID

Pers_ID

F_Name และ L_Name

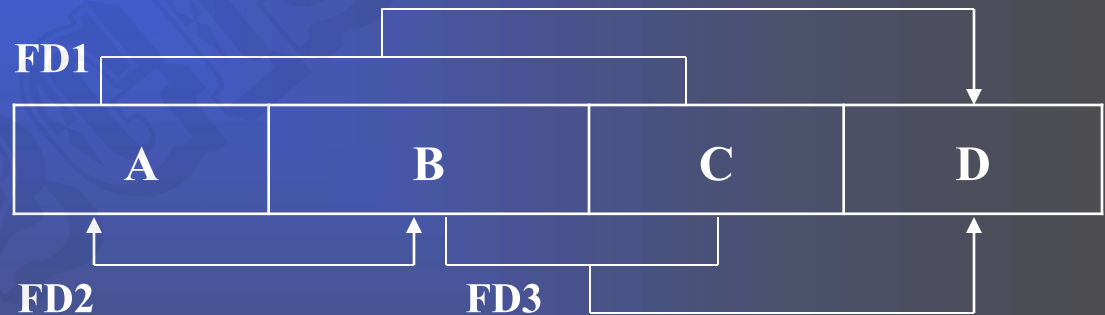
F_Name และ L_Name และ Date

F_Name และ L_Name และ Phone

จัดทำตารางข้อมูลให้มีคุณสมบัติอยู่ในNormal Form แบบบอยซ์คอดด์ หรือ BCNF (Boyce-Codd Normal Form)

ทุก Attribute ที่เป็น Determinant ในตารางต้องเป็นคีย์คู่แข่ง หรือ ไม่มี Attribute ที่ไม่ใช่ Key ที่สามารถระบุค่าของ Attribute ที่เป็น Primary key หรือ ส่วนใด ส่วนหนึ่งของ Primary key

(ข้อสังเกต ไม่มีการระบุว่า ตารางนั้นต้องมีคุณสมบัติ 3NF)

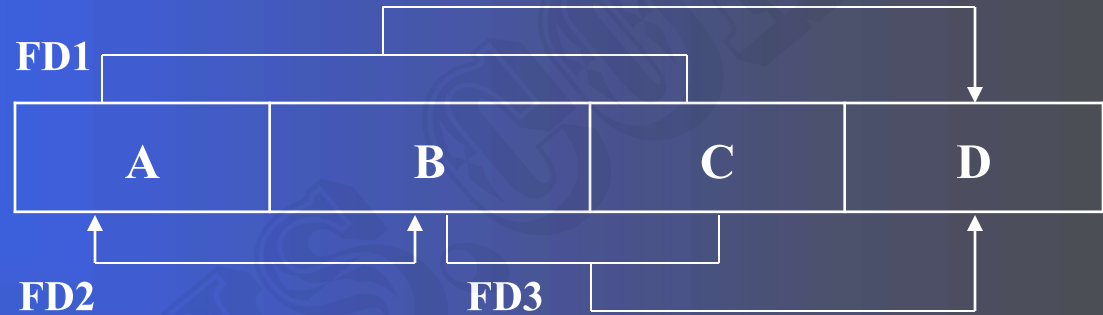
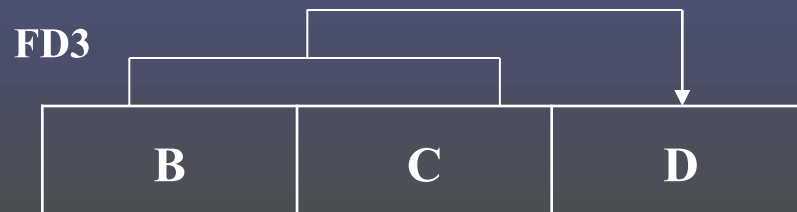
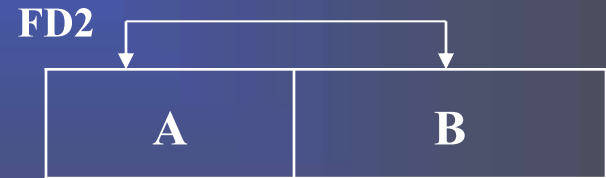
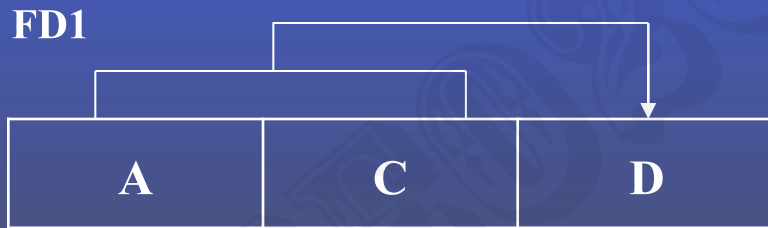
ตารางที่ไม่มีคุณสมบัติ BCNF

- คีย์คู่แข่งหลายคีย์
- คีย์คู่แข่งเป็นคีย์ผสม
- คีย์คู่แข่งมีบางส่วนเป็นตัวร่วม

FD1: A, C → D

FD2: A → B

FD3: B, C → D

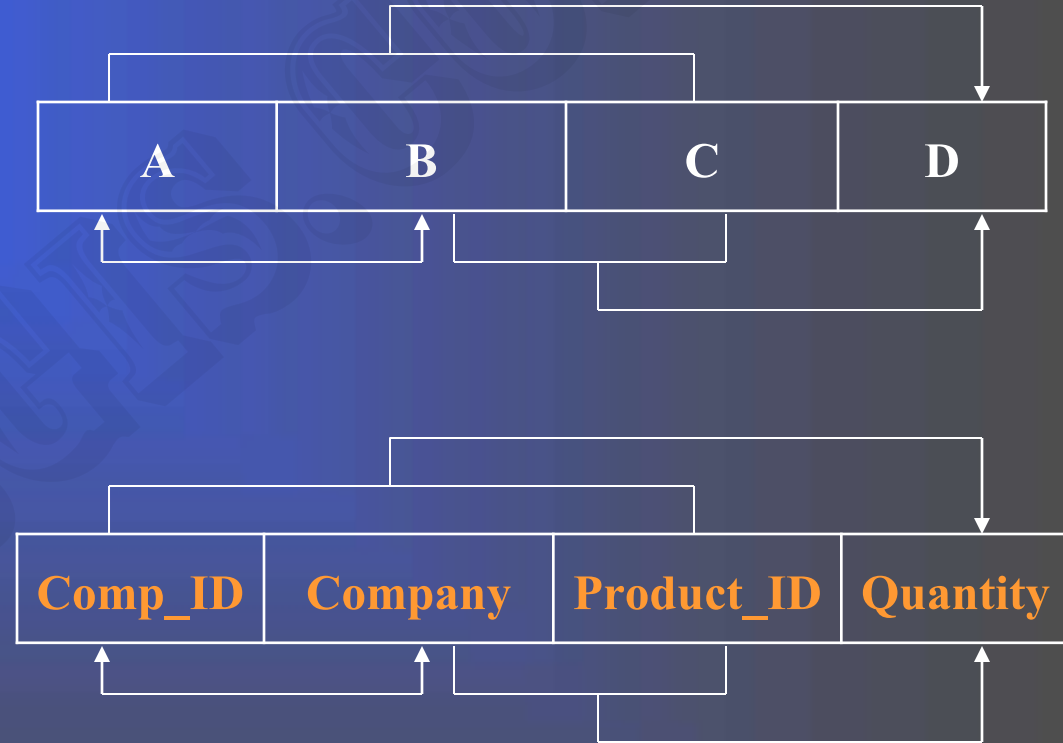
จัดทำตารางข้อมูลให้มีคุณสมบัติอยู่ในNormal Form แบบบอยซ์คอดด์ หรือ BCNF (Boyce-Codd Normal Form)ตารางที่ไม่มีคุณสมบัติ BCNFตารางที่มีคุณสมบัติ BCNF

จัดทำตารางข้อมูลให้มีคุณสมบัติอยู่ใน

Normal Form แบบบอยซ์คอดด์ หรือ BCNF (Boyce-Codd Normal Form)

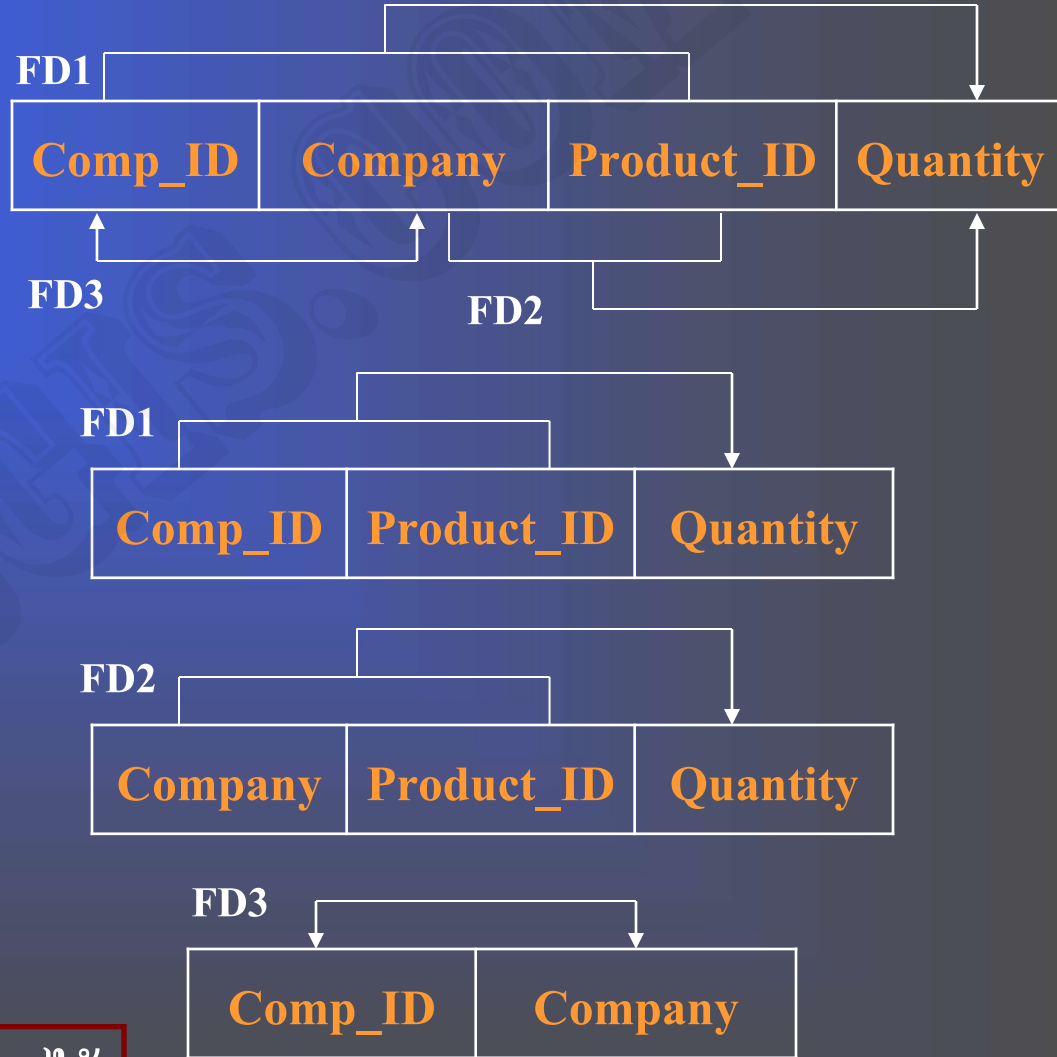
ตารางที่ไม่มีคุณสมบัติ BCNF

Comp_ID	Company	Product_ID	Quantity
C01	เอ เฟอร์นิเจอร์	PD01	50
C01	เอ เฟอร์นิเจอร์	PD02	20
C02	บี เฟอร์นิเจอร์	PD11	30
C02	บี เฟอร์นิเจอร์	PD12	25
C02	บี เฟอร์นิเจอร์	PD01	25
C03	C Furniture	PD02	30
C04	D Furniture	PD11	35
C04	D Furniture	PD03	40
C04	D Furniture	PD21	30



จัดทำตารางข้อมูลให้มีคุณสมบัติอยู่ในNormal Form แบบบอยซ์คอดด์ หรือ BCNF (Boyce-Codd Normal Form)ตารางที่ไม่มีคุณสมบัติ BCNF

Comp_ID	Company	Product_ID	Quantity
C01	เอ เฟอร์นิเจอร์	PD01	50
C01	เอ เฟอร์นิเจอร์	PD02	20
C02	บี เฟอร์นิเจอร์	PD11	30
C02	บี เฟอร์นิเจอร์	PD12	25
C02	บี เฟอร์นิเจอร์	PD01	25
C03	C Furniture	PD02	30
C04	D Furniture	PD11	35
C04	D Furniture	PD03	40
C04	D Furniture	PD21	30



FD1 มีค่าเหมือน FD2 จึงเลือกแค่ 1 ตารางได้

จัดทำตารางข้อมูลให้มีคุณสมบัติอยู่ในNormal Form ที่ 4 หรือ 4NFคุณสมบัติ

- ตารางนั้นต้องมีคุณสมบัติอยู่ใน BCNF
- ตารางนั้นต้อง**ไม่**มีการขึ้นต่อกันแบบเชิงกลุ่ม (MVD)

วิธีการ

ตรวจสอบว่ามีคุณสมบัติ 3NF หรือ BCNF หรือไม่

1. มีคุณสมบัติอยู่ใน 1NF (ไม่มีข้อมูลกลุ่มซ้ำ/ทุกช่องในตารางเป็นค่าเดียว)
2. มีคุณสมบัติอยู่ใน 2NF (ไม่มีการขึ้นต่อกันแบบบางส่วน)
3. มีคุณสมบัติอยู่ใน 3NF (ไม่มีการขึ้นต่อกันแบบ Transitive)
4. มีคุณสมบัติอยู่ใน BCNF (ทุก Determinant ในตารางต้องเป็นคีย์คู่แข่ง)

Emp_ID	Proj_ID	Language
E01	Po1	อังกฤษ
E01	Po1	จีน
E01	Po2	อังกฤษ
E01	Po2	จีน
E02	Po1	ฝรั่งเศส
E02	Po1	ญี่ปุ่น
E02	Po1	เยอรมัน
E03	Po3	ฝรั่งเศส

จัดทำตารางข้อมูลให้มีคุณสมบัติอยู่ในNormal Form ที่ 4 หรือ 4NFคุณสมบัติ

- ตารางนั้นต้องมีคุณสมบัติอยู่ใน BCNF
- ตารางนั้นต้อง**ไม่**มีการขึ้นต่อกันแบบเชิงกลุ่ม (MVD)

Emp_ID	Proj_ID	Language
E01	Po1	อังกฤษ
E01	Po1	จีน
E01	Po2	อังกฤษ
E01	Po2	จีน
E02	Po1	ฝรั่งเศส
E02	Po1	ญี่ปุ่น
E02	Po1	เยอรมัน
E03	Po3	ฝรั่งเศส

วิธีการ

ตรวจสอบว่ามีคุณสมบัติ 4NF โดยวิเคราะห์ Attribute ที่**มี**การขึ้นต่อกันแบบเชิงกลุ่ม (ความสัมพันธ์ที่เกิดขึ้นแบบ 1 : N)

Emp_ID $\longrightarrow$ Proj_ID

Emp_ID $\longrightarrow$ Language

Proj_ID และ Language ไม่มีความสัมพันธ์กัน

จัดทำตารางข้อมูลให้มีคุณสมบัติอยู่ในNormal Form ที่ 4 หรือ 4NFคุณสมบัติ

ตารางนั้นต้อง**ไม่มีการขึ้นต่อกันแบบเชิงกลุ่ม (MVD)**

วิธีการ

กำจัดการขึ้นต่อกันแบบเชิงกลุ่ม

Emp_ID	Language
E01	อังกฤษ
E01	จีน
E01	อังกฤษ
E02	ฝรั่งเศส
E02	ญี่ปุ่น
E02	เยอรมัน
E03	ฝรั่งเศส

Emp_ID	Proj_ID
E01	Po1
E01	Po2
E02	Po1
E03	Po3

Emp_ID	Proj_ID	Language
E01	Po1	อังกฤษ
E01	Po1	จีน
E01	Po2	อังกฤษ
E01	Po2	จีน
E02	Po1	ฝรั่งเศส
E02	Po1	ญี่ปุ่น
E02	Po1	เยอรมัน
E03	Po3	ฝรั่งเศส

ทั้ง 2 ตาราง
มีคุณสมบัติ
อยู่ใน 4NF

จัดทำตารางข้อมูลให้มีคุณสมบัติอยู่ในNormal Form ที่ 5 หรือ 5NFคุณสมบัติ

- ตารางนั้นไม่มีคุณสมบัติของการขึ้นต่อกันแบบจอยน์ (JD) หรือ
- ตารางนั้นต้องมีคุณสมบัติของการขึ้นต่อกันแบบจอยน์ (JD) และตารางที่แตกออกมาจะต้องมีคีย์คู่แข่งของตารางเดิมอยู่ด้วยเสมอ

Comp_ID	Company	City
C01	เอ เฟอร์นิเจอร์	กทม.
C02	บี เฟอร์นิเจอร์	เลย
C03	C Furniture	กทม.
C04	D Furniture	ตาก

Comp_ID	City
C01	กทม.
C02	เลย
C03	กทม.
C04	ตาก

Company	City
เอ เฟอร์นิเจอร์	กทม.
บี เฟอร์นิเจอร์	เลย
C Furniture	กทม.
D Furniture	ตาก

Comp_ID	Company	City
C01	เอ เฟอร์นิเจอร์	กทม.
C02	บี เฟอร์นิเจอร์	เลย
C03	C Furniture	กทม.
C04	D Furniture	ตาก

บทสรุปการทำ Normalization

Normalization

- เป็นกระบวนการลดความซ้ำซ้อนของข้อมูล
- เป็นกระบวนการแตกตารางเดิมให้เป็นตารางย่อย ๆ
- ตารางย่อย ๆ ต้องไม่มีข้อมูลสูญหายหรือข้อมูลเกินมาจากข้อมูลเดิม

การ Normalization มากเกินไป

ทั้ง 3 ตารางมี Primary key
ร่วมกัน มี Attribute ที่ไม่ใช่
key 1 ตัวแต่ไม่ผิดกฎเกณฑ์
แต่ไม่ควรแตกตารางย่อย ๆ
เพราะสามารถใช้ Primary
key ร่วมกันได้

Stu_ID	F_Name	L_Name	Credit
490101	อรุณ	สมบัติ	45
490102	วิชัย	เสรีไทย	77

Stu_ID	F_Name
490101	อรุณ
490102	วิชัย

Stu_ID	L_Name
490101	สมบัติ
490102	เสรีไทย

Stu_ID	Credit
490101	45
490102	77

บทสรุปการทำ Normalization

การย้ายไปใช้ Normal Form ที่ต่ำกว่า

การทำ Normalization ตั้งแต่ต้นจนถึง 5NF บางครั้งอาจย้ายไปใช้ 3NF เนื่องจากต้องเชื่อมโยงข้อมูลมากทำให้เสียเวลา มีผลกระทบต่อประสิทธิภาพการเรียกดูข้อมูล การนำข้อมูลมาพิมพ์เป็นรายงาน ต้องขึ้นอยู่กับนโยบายและลักษณะงานของหน่วยงาน

การออกแบบที่ดีโดยใช้ E-R diagram

ทำให้ผลลัพธ์ในรูปของตารางดีด้วย อีกทั้งยังทำให้ ตารางมีคุณสมบัติ 1NF ทุกครั้ง บ่อยครั้งพบว่ามีความสัมพันธ์ 3NF หรือ BCNF และมีความสัมพันธ์ 5NF โดยปริยาย (ถึงแม้ว่า ตารางมีความสัมพันธ์ 3NF หรือ BCNF แล้วก็ตามก็ต้องตรวจสอบคุณสมบัติ 5NF เสมอ)